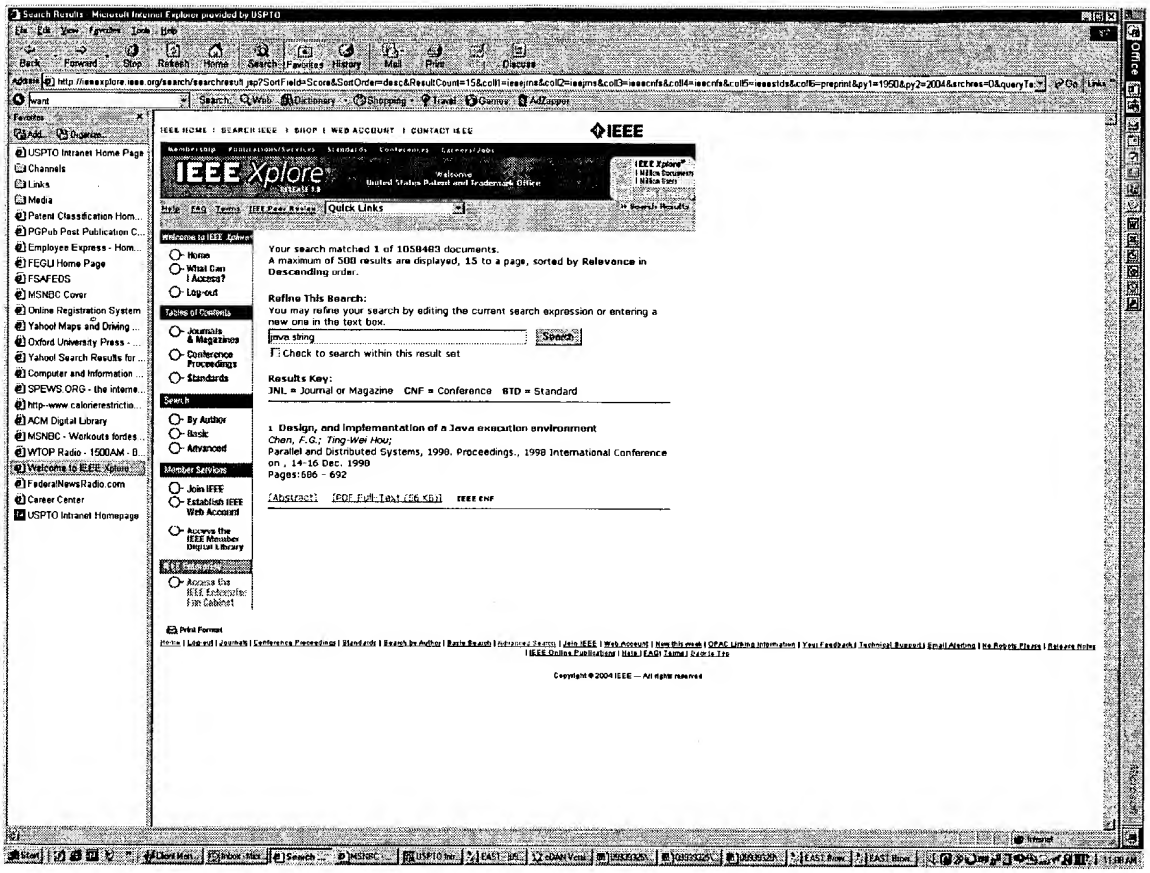
L Number	Hits	Search Text	DB	Time stamp
1	21587	string same character	USPAT; US-PGPUB	2004/08/03 14:08
2	3005	(one adj byte) same (two adj byte)	USPAT; US-PGPUB	2004/08/03 14:10
3	467	(string same character) and ((one adj byte) same (two adj byte))	USPAT; US-PGPUB	2004/08/03 14:09
4	0	(string with chatacter) same (one adj byte) same (two adj byte) same java	USPAT; US-PGPUB	2004/08/03 14:13
5	0	(string same chatacter) same (one adj byte) same (two adj byte) same java	USPAT; US-PGPUB	2004/08/03 14:13
6	0	(string same chatacter) same (one adj byte) same (two adj byte) and java	USPAT; US-PGPUB	2004/08/03 14:13
7	1	5664189.pn.	USPAT; US-PGPUB	2004/08/03 14:11
8	0	(string same chatacter) same ((one adj byte) or (two adj byte)) same java	USPAT; US-PGPUB	2004/08/03 14:13
9	3	(string with character) same (one adj byte) same (two adj byte) same java	USPAT; US-PGPUB	2004/08/03 14:14
10	7	(string same character) same (one adj byte) same (two adj byte) same java	USPAT; US-PGPUB	2004/08/03 16:20
13	10	("5303149" | "5367675" | "5488725" | "5548758" | "5717915" | "5732265" | "5813009" | "5826265" | "5829006" | "5838965").PN.	USPAT	2004/08/03 14:31
11	11	(string same character) same (one adj byte) same (two adj byte) and java	USPAT; US-PGPUB	2004/08/03 16:20
12	8	(string same character) same ((one adj byte) or (two adj byte)) same java	USPAT; US-PGPUB	2004/08/03 15:10
14	1	("6581077").PN.	USPAT; US-PGPUB	2004/08/03 15:13
15	1	("6557032").PN.	USPAT; US-PGPUB	2004/08/03 15:14
16	1	("6557023").PN.	USPAT; US-PGPUB	2004/08/03 16:09
17	0	("wo73894").PN.	EPO; JPO; DERWENT; IBM_TDB	2004/08/03 16:09
18	0	("wo73894").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/08/03 16:09
19	1	(string same character) same (one adj byte) same (two adj byte) same java	EPO; JPO; DERWENT; IBM_TDB	2004/08/03 16:20
20	1	(string same character) same (one adj byte) same (two adj byte) and java	EPO; JPO; DERWENT; IBM_TDB	2004/08/03 16:20
-	3	java adj string adj library	USPAT; US-PGPUB	2004/08/03 14:08

-	9	("5579518" | "5664189" | "5784069" | "5793381" | "5875335" | "5966702" | "6049869" | "6166666" | "6400287").PN.	USPAT	2004/08/02 13:37
-	10	("5440482" | "5485373" | "5758314" | "5787452" | "5793381" | "5832507" | "5873111" | "5929729" | "6049869" | "6055365").PN.	USPAT	2004/08/02 13:39
-	4	("5548507" | "5634134" | "5640587" | "5715466").PN.	USPAT	2004/08/02 13:40
-	3	("5444445" | "5784071" | "5793381").PN.	USPAT	2004/08/02 13:51
-	6	string with (respresentation or model) with byte with character	USPAT; US-PGPUB	2004/08/02 15:29
-	4	("5153729" | "5479609" | "5600821" | "6031964").PN.	USPAT	2004/08/02 14:19
-	48	java same (string adj object)	USPAT; US-PGPUB	2004/08/02 15:30
-	0	java same (string adj object) same bype same character	USPAT; US-PGPUB	2004/08/02 15:31
-	0	java same (string adj object) same bype	USPAT; US-PGPUB	2004/08/02 15:31
-	9	java same (string adj object) same byte	USPAT; US-PGPUB	2004/08/02 15:54
-	4	("5768564" | "6029207" | "6158044" | "6305009").PN.	USPAT	2004/08/02 15:39
-	16	("5187787" | "5475845" | "5504892" | "5627979" | "5692183" | "5790855" | "5878432" | "5915252" | "5924101" | "6016495" | "6018743" | "6061505" | "6061515" | "6125364" | "6141660" | "6173439").PN.	USPAT	2004/08/02 15:45
-	0	(string adj object) same respresentation same byte	USPAT; US-PGPUB	2004/08/02 15:55
-	7	(string adj object) same representation same byte	USPAT; US-PGPUB	2004/08/02 15:59
-	402	string same representation same byte	USPAT; US-PGPUB	2004/08/02 15:59
-	42	string same representation same (one adj byte)	USPAT; US-PGPUB	2004/08/02 16:00
-	1	string same representation same (one adj byte) same flag	USPAT; US-PGPUB	2004/08/02 16:00
-	8	string same representation same (one adj byte) same (two adj byte)	USPAT; US-PGPUB	2004/08/02 17:13
-	2	(("6708177") or ("6370581")).PN.	USPAT; US-PGPUB	2004/08/02 17:13
-	2	("6260083" | "6513002").PN.	USPAT	2004/08/02 17:15
-	4	6260083.URPN.	USPAT	2004/08/02 17:17
-	2	("6260083" | "6513002").PN.	USPAT	2004/08/02 17:17


Search Results - Microsoft Internet Explorer provided by USPTO
IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE
IEEE
IEEE Xplore
Quick Links
Your search matched 1 of 1058483 documents.
A maximum of 500 results are displayed, 15 to a page, sorted by Relevance in Descending order.
Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.
java string
Search
Check to search within this result set
Results Key:
JNL = Journal or Magazine CNF = Conference STD = Standard
1 Design, and implementation of a Java execution environment
Chan, F.G.; Ting-Wei Hou;
Parallel and Distributed Systems, 1990. Proceedings., 1990 International Conference on , 14-16 Dec. 1990
Pages:686 - 692
[Abstract] [PDF Full-Text (256 KB)] IEEE CNF
Copyright © 2004 IEEE — All rights reserved

Terms used **Java string**

Found **18,456** of **139,988**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 The Java programming language: Rethinking Java strings

Paolo Boldi, Sebastiano Vigna

June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java**

Full text available: pdf(55.14 KB) Additional Information: full citation, abstract

The Java string classes, String and StringBuffer, lie at the extremes of a spectrum (immutable, reference-based and mutable, content-based). Motivated by data-intensive text applications, we propose a new string class, MutableString, which tries to embody the best of both approaches.

2 Extending Java for high-level Web service construction

Aske Simon Christensen, Anders Møller, Michael I. Schwartzbach

November 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 6

Full text available: pdf(947.02 KB) Additional Information: full citation, abstract, references, citings, index terms

We incorporate innovations from the <bigwig> project into the Java language to provide high-level features for Web service programming. The resulting language, JWIG, contains an advanced session model and a flexible mechanism for dynamic construction of XML documents, in particular XHTML. To support program development we provide a suite of program analyses that at compile time verify for a given program that no runtime errors can occur while building documents or receiving form input, and ...

Keywords: Interactive Web services, XML, data-flow analysis

3 Using Java reflection to automate extension language parsing

Dale Parson

December 1999 **ACM SIGPLAN Notices , Proceedings of the 2nd conference on Domain-specific languages**, Volume 35 Issue 1

Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, index terms

An extension language is an interpreted programming language designed to be embedded in a domain-specific framework. The addition of domain-specific primitive operations to an embedded extension language transforms that vanilla extension language into a domain-specific language. The LUxWORKS processor simulator and debugger from Lucent uses Tcl as its extension language. After an overview of extension language embedding and LUxWORKS experience, this paper looks at using Java reflection and ...

4 Java and distributed object models: an analysis

Marjan Hericko, Matjaz B. Juric, Ales Zivkovic, Ivan Rozman, Tomaz Domajnko, Marjan Krisper
December 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 12

Full text available: pdf(871.07 KB) Additional Information: full citation, abstract, citings, index terms

Java has an important role in building distributed object oriented web enabled applications. In the article an analysis of two distributed object models in context of Java language is presented. Several aspects of RMI and CORBA such as features, maturity, support for legacy systems, learning curve and ease of development are compared. A special emphasis is given to the performances. Different testing scenarios give a complete overview about real world performances of both architectures. Based on ...

Keywords: CORBA, Java, RMI, distributed objects, performances

5 Targeting GNAT to the Java virtual machine
Cyrille Comar, Gary Dismukes, Franco Gasperoni
November 1997 **Proceedings of the conference on TRI-Ada '97**

Full text available: pdf(1.72 MB) Additional Information: full citation, references, citings, index terms

6 A comparison of the object-oriented features of Ada 95 and Java
Benjamin M. Brosgol
November 1997 **Proceedings of the conference on TRI-Ada '97**

Full text available: pdf(2.41 MB) Additional Information: full citation, references, citings, index terms

7 A comparison of the concurrency features of Ada 95 and Java
Benjamin M. Brosgol
November 1998 **ACM SIGAda Ada Letters , Proceedings of the 1998 annual ACM SIGAda international conference on Ada**, Volume XVIII Issue 6

Full text available: pdf(1.99 MB) Additional Information: full citation, references, citings, index terms

Keywords: Ada, Java, concurrency, inheritance anomaly, object-oriented programming, tasking, threads

8 Using Java APIs with native Ada compilers
Shayne Flint
November 1998 **ACM SIGAda Ada Letters , Proceedings of the 1998 annual ACM SIGAda international conference on Ada**, Volume XVIII Issue 6

Full text available: pdf(976.06 KB) Additional Information: full citation, references, index terms

Keywords: Ada, Java, Java native interface, bindings

9 Java 2 distributed object middleware performance analysis and optimization
Matjaz B. Juric, Ivan Rozman, Simon Nash
August 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 8

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, citings, index terms

This paper is focused on the performance analysis, comparison and optimization of distributed object middleware for Java 2: RMI (Remote Method Invocation), CORBA IDL (Interface Definition Language) and RMI-IIOP (Remote Method Invocation on Internet Inter-ORB Protocol). The paper presents the following contributions to the research on

distributed object performance. First, a detailed performance analysis is provided with the comparison. These results help to understand how the models perform. Sec ...

Keywords: CORBA, IDL, IIOP, Java, RMI, performance analysis and optimization

10 Interlanguage working without tears: blending SML with Java

Nick Benton, Andrew Kennedy

September 1999 **ACM SIGPLAN Notices , Proceedings of the fourth ACM SIGPLAN international conference on Functional programming**, Volume 34 Issue 9

Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms

A good foreign-language interface is crucial for the success of any modern programming language implementation. Although all serious compilers for functional languages have some facility for interlanguage working, these are often limited and awkward to use.This article describes the features for bidirectional interlanguage working with Java that are built into the latest version of the MLj compiler. Because the MLj foreign interface is to another high-level typed language which shares a garbage ...

11 Compact Java binaries for embedded systems

Derek Rayside, Evan Mamas, Erik Hons

November 1999 **Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(124.35 KB) Additional Information: full citation, abstract, references, citings, index terms

Embedded systems bring special purpose computing power to consumer electronics devices such as smartcards, CD players and pagers. Java is being aggressively targeted at such systems with initiatives such as the Java 2 Platform, Micro Edition, which introduces certain efficiency optimizations to the Java Virtual Machine. Code size reduction has been identified as an important future goal for ensuring Java's success on embedded systems [20]. However, limited processing power and timing constraints ...

12 Combining Ada 95, Java byte code, and the distributed systems annex

Brad Balfour

November 1997 **Proceedings of the conference on TRI-Ada '97**

Full text available: pdf(1.75 MB) Additional Information: full citation, index terms

13 Java power tools: model software for teaching object-oriented design

Richard Rasala, Jeff Raab, Viera K. Proulx

February 2001 **ACM SIGCSE Bulletin , Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education**, Volume 33 Issue 1

Full text available: pdf(59.00 KB) Additional Information: full citation, abstract, references, citings, index terms

The *Java Power Tools* or JPT is a Java toolkit designed to enable students to rapidly develop graphical user interfaces in freshman computer science programming projects. Because it is simple to create GUIs using JPT, students can focus on the more fundamental issues of computer science rather than on widget management. In a separate article[4], we will discuss with examples how the JPT can help freshman students to learn about the basics of algorithms, data structures, classes, and interf ...

14 An interface between Java and APL

Mike Symes

June 2000 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL-Berlin-2000 conference**, Volume 30 Issue 4

Full text available: pdf(946.86 KB) Additional Information: full citation, abstract, references, index terms

This paper discusses an interface between the Java and APL languages. It is in the form of a report on some technology that has been developed for SHARP APL, though there are no aspects of the technology that are particularly special to that dialect of APL.The interface (called, for the purposes of this paper, the "APL-Java Interface") is a general facility for allowing APL programs and Java programs to work together. The project was started because it occurred to us that there is now a substant ...

15 Java and Postgres95

Bill Binko

November 1996 **Linux Journal**

Full text available: html(22.88 KB) Additional Information: full citation, index terms

16 Visualising Java data structures as graphs

John Hamer

January 2004 **Proceedings of the sixth conference on Australian computing education - Volume 30**

Full text available: pdf(115.56 KB) Additional Information: full citation, abstract, references

We present a simple, general-purpose tool for visualising Java data structures. The tool uses Java reflection and an open-source graph drawing program to produce text-book quality depictions of arbitrary Java objects.The tool offers certain pedagogical advantages over other "heavy-weight" visualisation systems. Its simplicity and generality means that students are able to visualise their own data structures, rather than passively observing a limited range of "correct" visualisations prepared in ...

17 Simple problem solving in Java: a problem set framework

Viera K. Proulx, Richard Rasala, Jason Jay Rodrigues

May 2002 **The Journal of Computing in Small Colleges**, Volume 17 Issue 6

Full text available: pdf(65.88 KB) Additional Information: full citation, abstract, references, index terms

We present an application that allows for easy creation of simple problem solving exercises in Java, providing robust and safe I/O as well as a basic graphics window. We discuss possible uses for unit testing of classes and explore how the design of this application can be a case study in an object oriented design course.

18 JR: Flexible distributed programming in an extended Java

Aaron W. Keen, Tingjian Ge, Justin T. Maris, Ronald A. Olsson

May 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 3

Full text available: pdf(167.00 KB) Additional Information: full citation, abstract, references, index terms

Java provides a clean object-oriented programming model and allows for inherently system-independent programs. Unfortunately, Java has a limited concurrency model, providing only threads and remote method invocation (RMI).The JR programming language extends Java to provide a rich concurrency model, based on that of SR. JR provides dynamic remote virtual machine creation, dynamic remote object creation, remote method invocation, asynchronous communication, rendezvous, and dynamic process creation ...

Keywords: Concurrency, Java, SR, concurrent object-oriented programming

19 Data transfer between Java Applets and legacy APL systems

B. Amos, G. Disney, D. Sorrey

January 1998 **ACM SIGAPL APL Quote Quad , Proceedings of the conference on Share knowledge share success**, Volume 28 Issue 4

Full text available: pdf(843.73 KB) Additional Information: full citation, abstract, index terms

The rise of Internet technologies (particularly Java) provides many benefits for the

development and deployment of user interfaces. In many cases, however, the back end system is behind the times: Internet hostile, no object orientation, etc. How can data be transferred between the new generation front end and the old generation back end without compromising the strengths or integrity of either? This paper discusses the use of customised Java data serialisation to achieve this goal against a lar ...

20 Dynamic class loading in the Java virtual machine

Sheng Liang, Gilad Bracha

October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10

Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

Class loaders are a powerful mechanism for dynamically loading software components on the Java platform. They are unusual in supporting all of the following features: *laziness, type-safe linkage, user-defined extensibility*, and *multiple communicating namespaces*.We present the notion of class loaders and demonstrate some of their interesting uses. In addition, we discuss how to maintain type safety in the presence of user-defined dynamic class loading.

CiteSeer Find: Java string Documents Citations

Searching for **PHRASE java string**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date
Try: Google (CiteSeer) Google (Web) CSB DBLP
22 documents found. **Order: number of citations.**

Chava: Reverse Engineering and Tracking of Java Applets - Korn, Chen (1999) (Correct) (13 citations)
java.lang.**String java**.io.DataInputStream.readLine(Figure 1. A
java.lang.StringBuffer.StringBuffer(java.lang.**String) java**.lang.StringBuffer
java.lang.StringBuffer.append(double) java.lang.**String**
java.lang.StringBuffer.toString(void
www.research.att.com/~chen/chen/chava.ps

A Case for Language-Based Protection - Hawblitzel, von Eicken (1998) (Correct) (9 citations)
it is always type-safe to read characters from a **Java String** object, but this does not mean that one task
www.cs.cornell.edu/Info/People/hawblitz/TR98_1670/tr98_1670.ps

Support for Design Patterns through Graph Transformation Tools - Radermacher (1998) (Correct) (4 citations)
Package Classifier Intrinsic Language **String :java**"Transient Boolean :False Needs_update
INTERFACE Partition intrinsic partLanguage **string :Java**"cardinality string :1"Package
inf2-www.informatik.unibw-muenchen.de/People/ansgar/dito/papers/agtive99/agti\

Reverse Engineering of Java Applets - Korn, Chen, al. (1998) (Correct) (2 citations)
java.lang.**String java**.io.DataInputStream.readLine(Figure 1: A
void java.io.PrintStream.println(java.lang.**String) java**.lang.**String**
java.lang.StringBuffer.toString(
java.lang.**String java**.lang.StringBuffer.toString(void
www.cs.princeton.edu/~jlk/www8.ps

JaViz: A Client/Server Java Profiling Tool - Kazi, Jose, Ben-Hamida.. (2000) (Correct) (2 citations)
an inordinate amount of time is spent in certain **Java string** package methods. ffl Distributed applications.
www-mount.ee.umn.edu/~ihkazi/paper/prof_paper.ps.gz

Mutable Strings in Java: Design, Implementation and.. - Paolo Boldi Sebastiano (Correct)
Universit degli Studi di Milano Abstract The **Java string** classes, String and StringBuffer, lie at the
a wide range of applications. 1 Introduction The **Java string** classes, String and StringBuffer, lie at the
vigna.dsi.unimi.it/ftp/papers/MutableStrings.ps.gz

Compact Approximation of Lattice Functions with Applications.. - Boldi, Vigna (2002) (Correct)
of the project is a complete rewrite of the **Java string** classes [5] with better algorithms and
algorithms, 2 but often it backfires (as the **Java string** classes show)2 Notation For each natural
vigna.dsi.unimi.it/ftp/papers/CompactApproximators.ps.gz

Rethinking Java Strings - Boldi, Vigna (2003) (Correct)
Rethinking **Java Strings** Paolo Boldi Sebastiano Vigna Dipartimento di degli Studi di Milano, Italy Abstract The **Java string** classes, String and StringBuffer, lie at the
gongolo.usr.dsi.unimi.it/~vigna/ftp/papers/RethinkingJavaStrings.ps.gz

The Jungle Database Search Engine - Böhlen, Bukauskas, Dyreson (1999) (Correct)
will #nd all the tuples that contain the **string "Java"** in some attribute value. The user can then
www.cs.auc.dk/~linb/publications/sigmod1999.pdf

Experiments in the Retrieval of Unsegmented Japanese Text at the.. - McNamee (2001) (Correct)
of the evaluation our system did not use the **Java String** type internally (though the code has since
research.nii.ac.jp/ntcir/workshop/OnlineProceedings2/paul.pdf

Transparent Access to Legacy Data in Java - Olivier Gruber Ibm (Correct)
#avor# the get#set methods are simply taking a **Java string**#arrayas parameter. The case of the object
www.sun.ca/research/forest/UK.Ac.Gla.Dcs.PJW1.Olivier_Gruber2_pdf.pdf

JNI - C++ integration made easy - Gabrilovich, Finkelstein (Correct)
as resource acquisition" idiom [2]This way, a **Java string** is transparently converted on access into a
field (stringField'in 'obj'then convert the **Java string** representation /to Ctype

'const char *
www.cs.technion.ac.il/~gabr/papers/cuj_jni.pdf

Java to IDLLanguageMapping Specification - Newedition June Copyright (Correct)
.1-4 1.2.4.1 The **Java String** Type .1-5
{ public int getY(1.2.4.1 The **Java String** Type The java.lang.String class is a
www.omg.org/cgi-bin/doc?formal/99-07-59.ps

Language to IDLMapping - This Is Omg (Correct)
{public int getY(28.2.4.1 The **Java String** Type The java.lang.String class is a parameter type, return type, or data member, the **Java String** type is mapped to the type
ftp.omg.org/pub/docs/ptc/99-03-09.pdf

FAMIX Java language plug-in 1.0 - Author Sander Tichelaar (Correct)
Java models this attribute always contains the **string "Java"**SourceDialect The Java language doesn't
www.iam.unibe.ch/~famoos/FAMIX/Plugins/JavaPlugin1.0.ps.gz

Java interface programming: Everything off the best - Website Ve Ever (Correct)

(including strings and arrays)Accessing **Java Strings** Strings are a particularly useful kind of Java
functions to ease the task of handling **Java strings** in native code. The programmer can translate
www.cs.cmu.edu/~illah/HOWTO/javamanual.pdf

Data Types - Evett (Correct)
Unix's grep e.g: A-Za-z]A-Za-z\d]**Java String** class primitive String Length Encoding .
delimiter .Dynamic -SNOBOL4, Perl, Cand **Java String** classes Utility (of string types)Aid to
www.emunix.emich.edu/~evett/ProgrammingLanguages/LectureNotes/ProgLang5

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) CSB DBLP